REMARKS

Reconsideration of the present application is respectfully requested. Claims 1, 19 and 38 have been amended. Claims 53-55 have been canceled. Claims 56-68 are newly added. No new matter has been added.

Rejection under 35 U.S.C. § 112, First Paragraph / Objection to Applicants' Last Amendment

The Office objects to the addition of claims 53-55 in the last amendment (filed on May 16, 1005) and rejects those claims under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement (Office Action, pages 3-5). The Examiner contends that the subject matter of those claims is not supported by the original disclosure.

Note that claims 53-55 have been canceled; however, subject matter similar to that recited in claims 53, 54 and 55 has been incorporated into independent claims 1, 19, 38 and 56.

Applicants respectfully traverse the rejection. The Examiner cites Applicants' specification and states (Final Office Action, p. 4)(emphasis provided by Examiner):

> "The specification clearly states, 'the streamlets which are predicted to be needed at a given point during execution of the application are automatically sent to the client 14 so that they are generally present before the application attempts to access them. Both code and data, including external files used by the application, can be predictively streamed 20 [sic] in this manner and, the server is generally unconcerned with the actual content of the streamlets which are sent,' page 14, lines 12-15."

Applicants respectfully disagree with the Examiner's contention that the subject matter of claims 53-55 is unsupported by the original disclosure. The subject matter of

canceled claims 53-55 (and, correspondingly, the subject matter inserted into independent claims 1, 19, 38 and 56 in this amendment) is described clearly and extensively in numerous places throughout the original description. In fact, the passage from Applicants' specification cited by the Examiner (quoted above) is an example of support for the limitations in question, so Applicants find this rejection difficult to understand (note that the page and line numbers cited by the Examiner are incorrect; the cited text actually appears at page 12, lines 16-21 in Applicants' specification).

The following passages are additional examples of support in the specification for the subject matter of claims 53-55 (which has been incorporated into claims 1, 19, 38 and 56):

Specification, p. 7, lines 6-17 (emphasis added):

> The server system comprises an application streaming manager which coordinates the transmission of application streamlets to one or more client systems. To improve responsiveness of the application on the client side, the server uses a predictive engine to determine an optimal order to provide streamlets to a client. The predictive engine operates on one or more predictive models which can be generated using information gathered from user interaction with the streaming application and monitoring the order in which various sections of the application's files are accessed as the application runs. Several predictive models can be provided for a given application depending on the various ways in which the application can be used. Alternatively, a single predictive model can be provided with different flows to reflect various types of user behaviors and appropriate weightings between the flows provided. The predictive model can be further modified based on feedback provided by the client system about how an application is being run on a particular client system.

• • •

Specification, p. 11, line 21 – page 13, line 12(emphasis added):

> The server also has access to a predictive data model 172 which contains information about the usage patterns of the application, such as

the probability that the application will require a given streamlet block Bn if the last block used was Bm. A user database 173 can also be provided for storing various user-specific information for use in authorizing access to streamed application, storing information related to streaming order, etc. The sever system is discussed in more detail below.

The preferred client system 14 comprises an operating system 140 which can access a variety of local system resources 142, such as data storage devices and memory. A streaming support system module 102 is provided and contains streaming control software which is generally configured to initiate the streaming of streamlets from the server 12 through, e.g., network socket 144, initiate the execution of a specified streaming application 100, process received streamlets, request specific streamlets from the server when necessary, and make received streamlets available for use by the application in a manner which is generally transparent to the native operating system 140.

Preferably, the server 12 is configured to automatically forward sequences of streamlets to the client using a predictive streaming engine which selects streamlets to forward according to the predictive model, such as a dynamic statistical knowledge base generated by analyzing the various sequences in which the application program attempts to load itself into memory as various program features are accessed. As a result, the streamlets 18 which are predicted to be needed at a given point during execution of the application are automatically sent to the client 14 so that they are generally present before the application attempts to access them. Both code and data, including external files used by the application, can be predictively streamed in this manner and, the server is generally unconcerned with the actual content of the streamlets which are sent.

Various statistical and other predictive techniques can be used to analyze the sequence of code and data loads generated by an operating system as it executes an application and determine an optimal order to push the application streamlets to the client. In one embodiment, the predictive knowledge base can be used which can be viewed as a graph where a node is a user request (e.g. save, load) or a request by the operating system to load a given portion of one the application files, and an edge is the calculated probability that such a request will be made. A simple example of such a graph is shown in FIG. 3. For simplicity, the nodes are illustrated as basic application functions. By examining the links flowing from a given node, the predictive engine can easily determine the most likely future requests and, with access to an appropriate database, determine the streamlets which will be needed by the application to execute those requests. When the user strays from the predictive path and a streamlet fetch request is issued, the predictive routine can be re-centered on a node which corresponds to functionality associated with the requested streamlets and the predictive stream restarted from that point.

• • •

Specification, p. 14, lines 14-16)(emphasis added):

> The prediction engine 402 is accessed by the streaming manager 400 and identifies the next one or more streamlets which are predicted to be most appropriate to send to a given client at a given time.

• • •

Specification, p. 20, lines 6-10)(emphasis added):

> Once a streaming application has started, the prediction engine is queried to determine the next streamlet or streamlets which are appropriate to send to the client's system. (Step 600). The appropriate streamlets are then retrieved from the application library (step 602) and transmitted to the client. (Step 604).

The above are just a few examples of support in Applicants' specification for the subject matter in question. Applicants have not attempted to quote here every example of such support.

Therefore, Applicants respectfully submit that the rejection under 35 U.S.C. § 112, first paragraph, and the objection to Applicants' last amendment is clearly improper and should be withdrawn.

Rejection under 35 U.S.C. § 112, Second Paragraph

The Office rejected claims 53-55 under 35 U.S.C. § 112, second paragraph, as failing to provide sufficient antecedent basis for the phrase "the client" in those claims. This rejection is moot in view of the cancellation of claims 53-55.

Prior Art Rejection

Claims 1-9, 11, 12, 19-28, 30, 31, 38-43, 45 and 46 stand rejected under 35 U.S.C. § 103(a) based on U.S. Patent no. 5,892,915 of Duso et al. ("Duso") in view of

"Official Notice", U.S. Patent no. 6,219,671 of de Vries et al. ("de Vries"), and U.S. Patent no. 6,012,071 of Krishna et al. ("Krishna"). Applicants respectfully traverse the rejections.

The Examiner admits that Duso fails to disclose a prediction model stored within an application library (Office Action, p. 6). However, the Examiner takes "Official Notice" that "both the concept and advantages of providing a prediction model stored within the application library is well-known and expected in the art" (Final Office Action, p.6).

The Examiner further admits that Duso does not disclose that the software application has multiple files. However, the Examiner cites de Vries for such teaching (Final Office Action, pp. 7-8).

The Examiner further admits that Duso-deVries do not disclose that the application files and stream include executable code. However, the Examiner cites Krishna for such teaching (Final Office Action, p. 8).

Applicants respectfully submit that the Examiner's analysis of the cited art is incorrect. However, the rejection is believed to be moot in view of the above amendment to the independent claims.

Claim 1 now recites:

> 1. (Currently amended) A system for streaming a software application to a client comprising:
>
> an application library having stored therein a prediction model and application files of the software application, the application files including executable code;
>
> a streaming manager configured to send the application files to a client as a plurality of streamlets including executable code, each

streamlet corresponding to a particular data block in a respective application file; and

a streaming prediction engine configured to identify at least one streamlet which is predicted to be most appropriate to send to the client at a particular time, **based on an order in which code is predicted to be used when the software application is executed on the client,** in accordance with the prediction model. (Emphasis added.)

The cited references do not disclose or suggest, either individually or in combination, identifying at least one streamlet, which is predicted to be most appropriate to send to a client at a particular time, based on an order in which code is predicted to be used when the software application is executed on the client, in accordance with a prediction model. Hence, the cited references do not disclose all of the limitations of claim 1. For at least this reason, the subject matter of claim 1 cannot be considered obvious based on the cited references, whether they are taken individually or in combination.

In addition, in contrast with claim 1, the cited references also do not disclose or suggest a prediction model, nor a streaming prediction engine configured to identify at least one streamlet as most appropriate to send to a client at a particular time, in accordance with a prediction model. The Examiner incorrectly contends that Duso discloses these features at col. 18, lines 17-67, and Figure 28 (see Final Office Action, p. 6).

Duso discloses pre-fetching video data. Merely pre-fetching video data (or other multimedia data) cannot be considered the same as, or any suggestion of, identifying a streamlet of executable code as most appropriate to send to a client at a particular time, based on a prediction model. Streamable data files such as audio or video are

inherently organized sequentially, unlike software applications. Pre-fetching, as implemented in video streaming, simply relies on the well-defined sequence of playback to obtain the next required chunk of video data. That is not "prediction", much less identifying a streamlet as most appropriate to send to a client at a particular time based on a stored prediction model, per claim 1.

Hence, Duso does not disclose or suggest a streaming prediction engine or the use of a prediction model, as recited in Applicant's claims. Likewise, none of the other cited references disclose or suggest these features. For at least this additional reason, therefore, the cited references do not disclose all of the limitations of Applicants' independent claims. Consequently, the present invention cannot be considered obvious based on the cited references, whether they are taken individually or in combination.

In addition, while the Examiner admits that Duso fails to disclose a prediction model stored within an application library (Office Action, p. 6), the Examiner takes "Official Notice" that "both the concept and advantages of providing a prediction model stored within the application library is well-known and expected in the art" (Final Office Action, p.6). The Examiner cites various references to support this contention (Final Office Action, pp. 6-7).

The Examiner appears to cite these references mainly for the proposition that it would be obvious to store code and/or data in an application library. Applicants respectfully submit that the Examiner seems to miss the point. The issue is not whether it would be obvious to store code in an application library, generally. What is more relevant is whether it would be obvious to provide a prediction model in the first place,

where the prediction model is used in the specific manner and for the specific purpose recited in Applicants' claims, and to store such a prediction in an application library with application files including executable code, as claimed. The cited references do not support the conclusion that it would be obvious to do so. For at least this additional reason, therefore, the present invention has patentable merit over the cited references.

Independent claims 19 and 38 include limitations similar to those emphasized above regarding claim 1. Therefore, claims 19 and 38 are also patentable along with their dependent claims for similar reasons.

Similarly, regarding new independent claim 56, the cited references do not disclose or suggest generating a predictive model for use in determining blocks likely to be loaded by the software application at various points during execution of the software application. Furthermore, the cited references do not disclose or suggest generating an application package that includes the predictive model and a repository representing blocks of a software application, the application package for use by a streaming server to stream the software application to a client. Therefore, claim 56 is also patentable along with its dependent claims.

Dependent Claims

In view of the above remarks, a specific discussion of the dependent claims is considered to be unnecessary. Therefore, Applicants' silence regarding any dependent claim is not to be interpreted as agreement with, or acquiescence to, the rejection of such claim or as waiving any argument regarding that claim.

Conclusion

For the foregoing reasons, the present application is believed to be in condition for allowance, and such action is earnestly requested.

If any additional fee is required, please charge Deposit Account No. 02-2666.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: 11/10/05

Jordan M. Becker
Reg. No. 39,602

Customer No. 08791
12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1030
(408) 720-8300